Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Ajda Lampe

Semantična segmentacija slik za razpoznavanje notranjih prostorov

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Matej Kristan

Ljubljana, 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Razpoznavanje notranjih prostorov na podlagi slik je bistvenega pomena za številne sisteme, ki segajo od učinkovitega iskanja po slikovnih podatkovnih zbirkah do domačih robotov pomočnikov. V zadnjem času so se holistični pristopi izkazali za zelo uspešne, vendar ti pristopi niso sposobni pravilne kategorizacije v primerih, ko slikovno polje obsega več kot en prostor. V diplomski nalogi zato obdelajte problem razpoznavanja prostorov s postopki, ki temeljijo na semantični segmentaciji. Izdelajte lasten postopek za razpoznavanje, ki bo sposoben razpoznavanja na nivoju slikovnega elementa in lokalizacije prostora v sliki. Postopek ovrednotite na primerni podatkovni zbirki.

Iskreno se zahvaljujem mentorju doc. dr. Mateju Kristanu za usmerjanje, nasvete, pomoč ter veliko mero potrpežljivosti pri izdelavi diplomske naloge. Domnu Taberniku, Roku Mandeljcu, Petru Uršiču in Alanu Lukežiču sem zelo hvaležna za vso tehnično pomoč in moralno podporo.

Zahvaljujem se tudi svoji družini in Anžetu za vso podporo, ki sem jo večkrat zelo potrebovala tekom celotnega študija in za vzpodbudo, kadar sem sama izgubila zaupanje vase.

Na koncu se želim zahvaliti še študijskim kolegom, ki so mi polepšali študijska leta in mi pogosto dajali motivacijo za nadaljevanje.

Svojim najbližjim.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
ANN	artificial neural network	umetna nevronska mreža
CNN	convolutional neural network	konvolucijska nevronska mreža
LSI	linear shift-invariant	linearno neodvisen od premika
ReLU	rectified linear unit	izboljšana linearna enota
mIOU	mean intersection-over-union	povprečna vrednost količnika med presekom in unijo
FC	fully connected	polno povezan

Povzetek

Naslov: Semantična segmentacija slik za razpoznavanje notranjih prostorov

Razpoznavanje prostorov je zanimiv problem na področju računalniškega vida, ki je praktično uporaben na mnogo področjih v vsakdanjem življenju. Z razvojem mobilne robotike bo potreba po učinkovitem in točnem razpoznavanju prostorov rasla. V zadnjem času metode za klasifikacijo prostorov dosegajo vedno boljše rezultate z uporabo konvolucijskih nevronskih mrež, naučenih na veliki količini podatkov, vendar večina metod temelji na razpoznavanju celotne slike. Slabost teh sistemov se pokaže, kadar se na sliki pojavi več kot en prostor. V diplomskem delu smo razvili metodo, ki slabost obstoječih metod rešuje s semantično segmentacijo, pri tem pa smo se osredotočili na osem najpogostejših kategorij notranjih prostorov. Z uporabo dopolnjene in predelane zbirke podatkov smo izdelali in naučili tri konvolucijske nevronske mreže, ki se med seboj razlikujejo v številu polno povezanih nivojev. Njihovo točnost segmentacije in pravilnost detekcije smo numerično ovrednotili in vrednosti primerjali z rezultati obstoječe klasifikacijske mreže, ki dosega odlične rezultate pri klasifikaciji na nivoju slike. Rezultate mrež smo analizirali tudi kvalitativno. Naučene mreže presegajo rezultate referenčne, trenutno najboljše, metode za slabih 40% pri lokalizaciji prostora in za 20% pri detekciji objektov v sliki.

Ključne besede: računalniški vid, razpoznavanje prostorov, semantična segmentacija, konvolucijske nevronske mreže.

Abstract

Title: Semantic segmentation of images for indoor place recognition

Space recognition is an interesting computer vision problem with many practical applications. Improvements in field of mobile robotics will most likely increase the need for efficient and accurate scene recognition systems. Lately, room classification methods have reached high classification accuracy with the use of popular convolutional neural networks, trained on large datasets, but most of the methods are based on holistic classification. Their disadvantage shows when presented with an image of multiple places. In this thesis we present a method that addresses the disadvantage of existing methods by use of semantic segmentation. In the work we focus on recognizing 8 most common indoor place categories. We improved and changed an existing dataset according to the problem and used it to build and train three convolutional neural networks with different numbers of fully-connected layers. We evaluated their segmentation and detection accuracy with use of mean intersection-over-union measure and F-measure, respectively, then compared obtained results with those of an existing holistic classification network, which achieves state-of-the-art results on the task of image-level classification. We also give a qualitative analysis of trained networks' results. Results show that our method outperforms the current state-of-the-art method by almost 40% on the task of place localization and by 20% on the task of place recognition.

Keywords: computer vision, place recognition, semantic segmentation, convolutional neural networks.

Poglavje 1

Uvod

Problem razpoznavanja prostorov je zanimiv problem na področju računalniškega vida, ki je v praksi uporaben na mnogih področjih. Sistem za razpoznavanje zunanjih prostorov v avtonomnih vozilih bi lahko prilagajal način vožnje glede na vrsto okolja v katerem se giblje, avtonomna plovila pa bi lahko samodejno ugotovila kdaj se nahajajo v pristanišču. Hišni robot, ki bi bil sposoben razpoznavati prostore v stanovanju, ne bi potreboval mape, poleg tega pa bi lahko avtomatsko vršil akcije glede na to v katerem prostoru se nahaja. Sistem za razpoznavanje prostorov je lahko uporaben tudi na internetnih straneh za razvrščanje slik glede na prostore v njih, na primer spletne strani nepremičninskih agencij, fotografije na družabnih omrežjih in podobno.

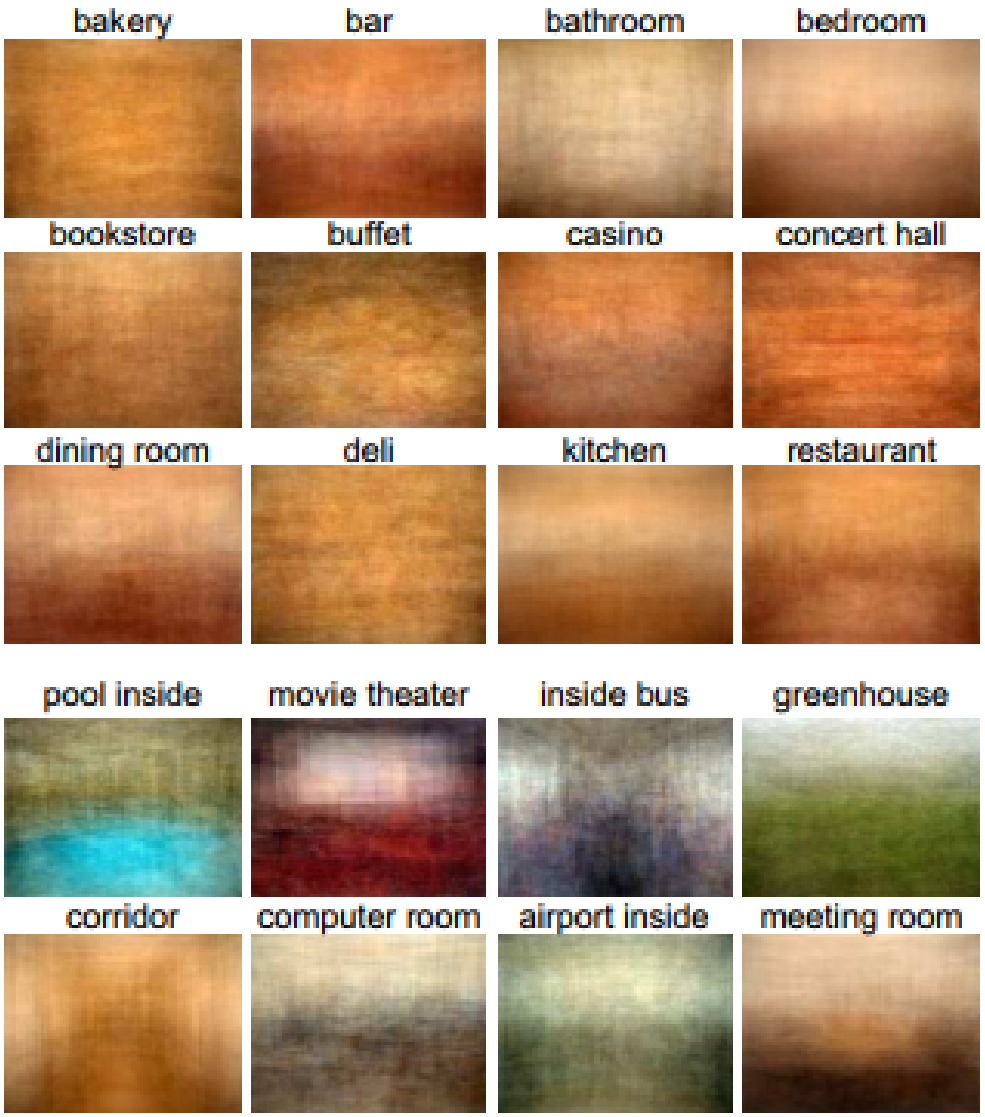
Razpoznavanje prostorov temelji na iskanju značilnic v slikah, na podlagi katerih sistem lahko uvrsti sliko v nek razred. Problem je zahteven, saj so si nekatere kategorije med seboj zelo podobne ali celo delno sovpadajo, na primer spalnica in otroška soba (Slika 1.1), hkrati pa so slike znotraj iste kategorije lahko zelo različne (Slika 1.2). Najpogosteje sistemi, ki dobro razpozvanajo zunanja prizorišča, odpovejo pri razpoznavanju notranjih prostorov [1]. Razlog je različnost lastnosti, ki opisujejo določeno kategorijo in na podlagi katerih jo lahko ločimo od ostalih kategorij. Nekateri prostori, npr. hodnik, so lažje predstavljivi s pomočjo globalnih prostorskih lastno-

sti, drugi, npr. kuhinja, pa z objekti, ki se v njih najpogosteje pojavljajo. Zahtevnost problema je mogoče razbrati iz srednjih slik (angl. *mean image*) posameznih prostorov, prikazanih na Sliki 1.3.

Slika 1.1: Meja med otroško sobo in spalnico ni jasna. Slika vzeta iz podatkovne zbirke MIT Indoor [1].

Slika 1.2: Slike dnevnih sob so med seboj lahko zelo različne. Sliki sta del podatkovne zbirke MIT Indoor [1].

Veliko obstoječih metod temelji na prepoznavanju objektov, ki sestavljajo prostore, ter na podlagi tega znanja klasificirajo celotno sliko v določeno kategorijo. Slabost teh sistemov se pokaže pri označevanju slik, ki vsebujejo



Slika 1.3: Povprečne slike nekaterih prostorov, vsebovanih v zbirki MIT Indoor. Posamezne kategorije pogosto ne vsebujejo izstopajočih značilnosti, ki bi jih razlikovale od drugih kategorij, zato jih je težko razločevati. Slika povzeta po [1].

več kot en prostor (Slika 1.4). Ponavadi ti modeli sicer pravilno napovejo enega izmed prostorov, vendar s tem zavržejo informacijo o ostalih prostorih,

ki se potencialno pojavijo v sliki. V tem poglavju je opisano dosedanje delo na omenjenem področju, predstavitev prispevkov diplomske naloge ter njena okvirna zgradba.

Slika 1.4: Slike, na katerih se pojavlja več različnih prostorov. Slike so del podatkovne zbirke, uporabljene v tej diplomski nalogi.

1.1 Pregled področja

Večina rešitev na področju razpoznavanja prostorov uporablja holistične deskriptorje. Mnogo pristopov temelji na kodiranju slike v vektor značilk in na podlagi tega klasificira sliko. En način kodiranja je vreča besed (angl. *bag of words*, BoW), metode ki ga uporabljajo pa so številne. V [2] je opisana predstavitev z afino invariantnimi deskriptorji, klasifikacija pa je izvedena v enem primeru z naivnim Bayesovim klasifikatorjem, v drugem pa z metodo podpornih vektorjev (angl. *support vector machines*). Članek [3] temelji na iskanju najznačilnejših delov, opisanih s histogramom orientiranih gradientov (angl. *histogram of oriented gradients*) in njihovo uporabo v holističnem deskriptorju vreče besed oziroma delov. Tretji pristop, opisan v [4] uporablja predstavitev s histogrami značilk SIFT (angl. *Scale-invariant feature transform*) na podlagi informacije pridobljene z linearno diskriminantno analizo (angl. *linear discriminant analysis*).

Druga skupina holističnih pristopov temelji na detekciji objektov in klasifikacijo slike glede na pojavitev objektov. Avtorji članka [5] predstavijo

metodo z velikim številom vnaprej naučenih detektorjev objektov, odzive, ki jih vrnejo detektorji, pa uporabijo kot vhod v preprost klasifikator.

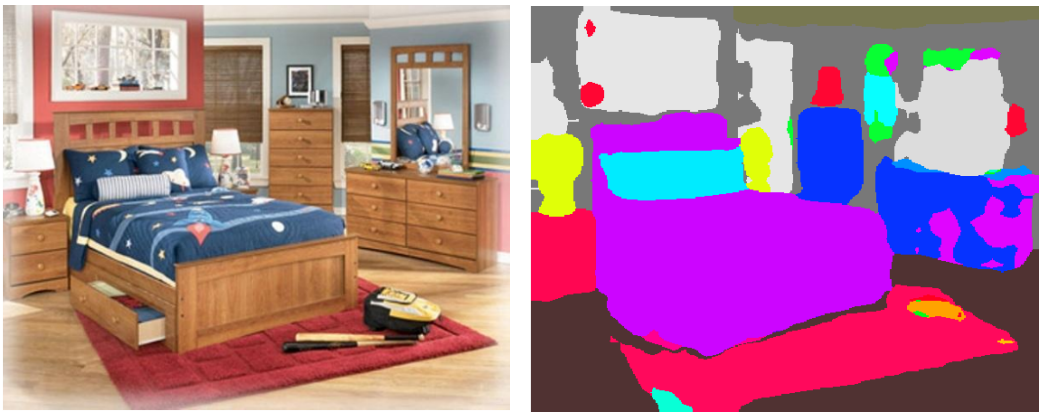
Pomembno točko v razvoju področja predstavlja pojavitev večjih podatkovnih zbirk, specializiranih za razpoznavanje prostorov. Med te zbirke sodijo zbirke MIT Indoor [1] s 67 kategorijami notranjih prostorov, SUN [6] s skupno 899 kategorijami, izmed katerih jih 397 vsebuje vsaj 100 primerov, Places205 [7] z 205 kategorijami ter Places365 [8], ki vsebuje več kot 400 razredov z najmanj 5000 učnimi primeri. Takšne zbirke med drugim zagotovijo lažjo primerljivost metod, ki so naučene in testirane na istih podatkih.

Lažja dostopnost večjih količin podatkov je botrovala vse pogostejši uporabi konvolucijskih nevronskih mrež. Pogosto se uporabljajo tudi modeli, naučeni na velikih splošnih zbirkah kot je ImageNet [9], nato pa priučeni na manjši specializirani zbirki. Primer je mreža PlacesNet [7], ki je naučena na zbirki Places205 razviti v okviru istega članka. V članku [10] so vhodi v konvolucijsko mrežo izseki iz slike na različnih skalah, odzivi pa so nato združeni. V [11] je sistem, ki generira aktivacijske mape po razredih. Te imajo visoko vrednost na mestih, kjer je največja verjetnost prisotnosti razreda, zato je metoda sposobna dokaj natančno lokalizirati nek razred kljub učenju na podatkovni zbirki z oznakami na nivoju slike.

Kot nasprotje holističnemu pristopu nekatere metode temeljijo na delih (angl. *part-based*), ki ponavadi bolje deljujejo na podatkih, kjer je del slike prekrit. Model, razvit v [1] temelji na prototipih, podobnih zvezdnim konstelacijam, ki združujejo globalno in lokalno informacijo o sliki. V [12] je opisan model, ki naključno vzorči dele in z uporabo regularizacije izbere optimalno podmnožico. Pristop, ki sliko predstavi z nerazvrščenimi deli, pridobljenimi s predlogi regij (angl. *region proposal*) in kodiranimi z uporabo konvolucijske nevronske mreže, je opisan v [13].

1.2 Prispevki

Glavni prispevek diplomske naloge je nov pristop za razpoznavanje prostorov na nivoju slikovnih točk. Tak sistem je sposoben razpoznati vse kategorije, ki se pojavijo v sliki in jih zna tudi lokalizirati. Razviti pristop temelji na uporabi konvolucijskih nevronskih mrež, ki se v zadnjem času vse pogosteje uporabljajo na raznih področjih računalniškega vida zaradi velike uspešnosti pri reševanju različnih nalog. Postopek, uporabljen v nalogi, s katerim sliko razdelimo na semantično povezane skupine slikovnih točk, kot prikazuje Slika 1.5, se imenuje semantična segmentacija in se do sedaj še ni uporabljal za namen razpoznavanja prostorov.



Slika 1.5: Semantična segmentacija je postopek razdelitve slike na enote, ki pripadajo istemu razredu. Slike so rezultat segmentacije na spletni predstavitvi sistema za segmentacijo scen [14].

Drugi prispevek naloge je izdelava anotirane podatkovne zbirke za razpoznavanje slik z večimi kategorijami prostorov. Kot osnova služi obstoječa zbirka za razpoznavanje prostorov, ki smo jo ustrezno anotirali in dopolnili z novimi primeri.

1.3 Zgradba diplomske naloge

Diplomska naloga je razdeljena na pet poglavij. V Poglavju 2 so opisane lastnosti in delovanje umetnih nevronskih mrež, ki so eden izmed pogosteje uporabljanih modelov na področju strojnega učenja in so služile kot osnova za razvoj konvolucijskih nevronskih mrež. Sledijo osnove konvolucije ter opis tistih lastnosti konvolucijskih nevronskih mrež, ki se razlikujejo od osnovnih umetnih nevronskih mrež ter tipi nivojev, ki se pojavljajo v konvolucijskih mrežah.

Poglavje 3 opisuje obstoječo programsko opremo, uporabljeno v diplomski nalogi in naš pristop k razpoznavanju prostorov. Prvi del je namenjen predstavitvi programskega ogrodja Caffe [15], drugi del pa predstavi mrežo Deeplab [16], ki je uporabljena kot osnova za učenje modela. Na koncu so opisane naše predelave mreže Deeplab za razpoznavanje prostorov.

Podatkovna zbirka in metode vrednotenja uspešnosti mreže so predstavljene v Poglavju 4, sledi pa jim predstavitev in vrednotenje dobljenih rezultatov. V Poglavju 5 so strnjene ugotovitve in izpostavljene možnosti za izboljšavo.

Poglavje 2

Nevronske mreže

Konvolucijske nevronske mreže (angl. *convolutional neural networks*, CNN) so nadgradnja umetnih nevronskih mrež (angl. *artificial neural networks*, ANN), zato je poznavanje osnovnega delovanja ANN potrebno za razumevanje CNN. Lastnosti, opisane v sledečih poglavjih, se nanašajo na usmerjene nevronske mreže (angl. *feedforward neural networks*), ki so služile kot osnova za razvoj CNN.

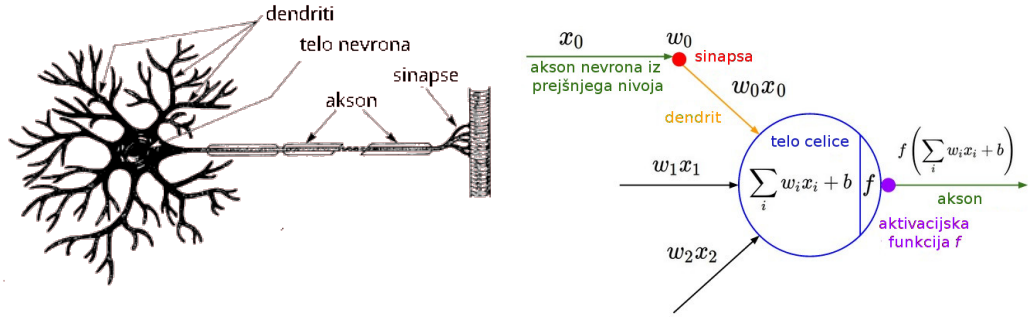
2.1 Umetne nevronske mreže

Nevronske mreže so računski model v strojnem učenju, ki s svojim delovanjem poskuša posnemati biološki centralni živčni sistem. Biološki nevron, osnovna enota živčnega sistema, je v grobem sestavljen iz dendritov, telesa, in aksona, kot prikazuje Slika 2.1a. Telo nevrona preko dendritov prejema elektro-kemične dražljaje iz drugih nevronov in jih, če so v kratkem času dovolj pogosti, posreduje preko aksona, ki se v sinapsi stika z dendriti naslednjega nevrona [17]. Nevroni se tako povezujejo v zapleteno mrežo. Z učenjem lahko nastajajo nove sinapse, obstoječe pa se krepijo ali slabijo.

Podobno pri računskem modelu vhodni signal x_0 do delesa nevrona pride preko sinapse, njena utež w_0 pa uravnava vpliv tega signala. Telo celice nato izračuna uteženo vsoto vseh vhodnih vrednosti $\sum_i w_i x_i + b$ in rezultat z

aktivacijsko funkcijo $f(\cdot)$ preslika na želeni interval (npr. med 0 in 1), izhod pa posreduje preko aksona naslednjemu nevronu. Aktivacijska funkcija v računskem modelu simulira pogostost proženja (angl. *firing rate*) v biološkem nevronu.

Nevroni v telesu živega bitja tvorijo zapleteno mrežo, nevroni v usmerjeni nevronski mreži, izmed katerih je napogosteje uporabljana arhitektura večnivojski perceptron (angl. *multilayer perceptron*, MLP), pa so organizirani v sloje, ki so med seboj povezani z enosmernimi povezavami, in ne vsebujejo ciklov. Ta lastnost nevronskih mrež močno poenostavi postopek vzvratnega razširjanja napake, ki je opisan nekoliko kasneje v tem poglavju. Sinapse v živalskem telesu ne predstavljajo samo ene uteži, ampak kompleksen nelinearen dinamični sistem, v katerem je pomemben natančen časovni razmak med zaporednimi signali, zaradi česar je umetna nevronska mreža samo zelo poenostavljen približek dejanskega živčnega sistema [18].



(a) Osnovni gradniki nevrona v človeškem telesu.

(b) Model nevrona v umetni nevronski mreži in primerjava gradnikov z biološkim nevronom. Slika povzeta po [18]

.

Slika 2.1: Primerjava biološkega nevrona in nevrona v umetni nevronski mreži.

Nevronske mreže so sestavljene iz vhodnega sloja, izhodnega sloja in poljubnega števila skritih slojev med njima. Število nevronov v vhodnem sloju je enako številu vhodnih atributov, v izhodnem pa številu razredov. Skriti sloji imajo lahko poljubno število nevronov.

2.1.1 Aktivacijska funkcija

Nevronsko mrežo brez aktivacijske funkcije $f(x)$, oziroma s funkcijo enako identiteti, lahko enačimo z linearnim klasifikatorjem. Uporaba aktivacijske funkcije je nujno potrebna, da izkoristimo poln potencial nevronske mreže, saj z uvedbo nelinearnosti lahko aproksimiramo zelo zapletene hiperravnine, ki ločujejo razrede podatkov. Vhod v aktivacijsko funkcijo je utežena vsota vhodnih aktivacij. Zaželeno je, da je aktivacijska funkcija nelinearna, in monotona, zanima pa nas tudi njena zaloga vrednosti [19]. Nujna lastnost vsake aktivacijske funkcije, ki se uporablja v ANN je zvezna odvedljivost, saj učenje mreže temelji na principu spusta po gradientu (angl. *gradient descent*) .

Sigmoidna funkcija realne vrednosti, ki so rezultat utežene vsote, preslika na interval med 0 in 1. Formula sigmoide $\sigma(x)$, ki je poseben primer logistične funkcije, je

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2.1}$$

kjer x predstavlja vhod v funkcijo. V preteklosti je bila pogosto uporabljana, ker jo je mogoče interpretirati kot pogostost proženja (angl. *firing rate*) nevrona. Poleg tega pa ima tudi lepo in pomembno lastnost, da je njen odvod mogoče izraziti kar s funkcijo samo, kot vidimo v spodnji formuli

$$\frac{\mathrm{d}}{\mathrm{d}x}\sigma(x) = \sigma(x)(1 - \sigma(x)). \tag{2.2}$$

Slabost sigmoidne funkcije je, da je pri vhodih z veliko absolutno vrednostjo, kjer se vrednost funkcije približuje 0 ali 1, lokalni, posledično pa tudi skupni, gradient skoraj enak 0, kot je razvidno iz Slike 2.2. Zato je pri uporabi sigmoidne funkcije potrebna skrbna inicializacija uteži, saj prevelike uteži zaradi uničevanja gradientov močno upočasnjujejo proces učenja.

Ker je zaloga vrednosti sigmoidne funkcije omejena na interval od 0 do 1, so vhodne vrednosti v naslednji nivo vse pozitivne. Zato je v nekaterih primerih primernejša uporaba **hiperboličniga tangensa**, izhodi katerega so

centrirani glede na izhodišče. Formuli hiperboličnega tangensa in njegovega odvoda sta

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}, \tag{2.3}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} \tanh(x) = 1 - \tanh^2(x). \tag{2.4}$$

Hiperbolični tangens lahko izrazimo tudi kot premaknjeno in skalirano sigmoidno funckijo, kot je razvidno iz spodnje enačbe:

$$\tanh(x) = 2\sigma(2x) - 1. \tag{2.5}$$

Hiperbolični tangens, tako kot sigmoidna funkcija, lahko povzroči uničevanje gradientov pri velikih vrednostih uteži, zato je pomembno kako so te inicializirane.

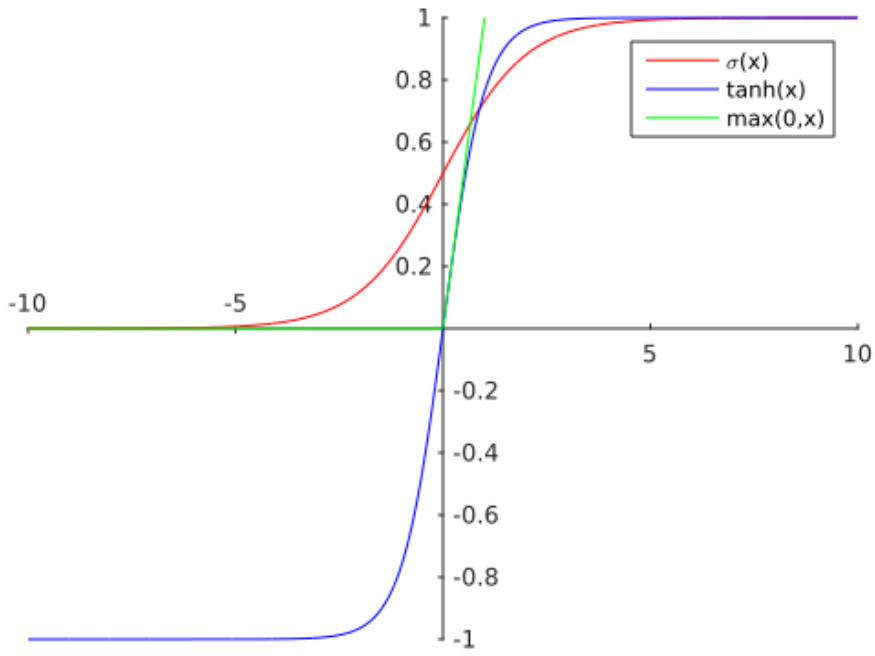
Enota ReLU implementira matematično funkcijo, ki negativne vhodne vrednosti preslika v nič, pozitivne pa ohrani. Njena formula in odvod sta

$$f(x) = \max(0,x), \tag{2.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \begin{cases} 1, & \text{če je } x > 0 \\ 0, & \text{sicer.} \end{cases} \tag{2.7}$$

V primerjavi s sigmoidno funkcijo in hiperboličnim tangensom funkcija ReLU pospeši konvergenco pri stohastičnem spustu po gradientu (Poglavje 2.1.2). Njena druga prednost pred prej omenjenima funkcijama je dejstvo, da je upragovanje pri ničli (angl. *zero thresholding*) veliko enostavnejša in hitrejša operacija. Vendar se pri velikih gradientih lahko uteži posodobijo tako, da se enota nikoli ponovno ne aktivira in so vsi naslednji gradienti enaki nič, kar imenujemo "umiranje"enot (angl. *dying ReLU*). Verjetnost tega pojava je manjša, če hitrost učenja (learning rate) ni previsoka.

Obstajajo tudi variacije ReLU enot, ki preprečijo "umiranje"enot, vendar imajo tudi te svoje slabosti, zato se izbira aktivacijske funkcije razlikuje od problema do problema.



Slika 2.2: Grafi sigmoidne funkcije (rdeča), hiperboličnega tangensa (modra) in ReLU funkcije (zelena).

2.1.2 Vzvratni prehod

Na podlagi izračunane napake pri klasifikaciji je v učni fazi potrebno prilagoditi uteži v nevronih, da bo napaka v naslednji iteraciji manjša. Napako je potrebno razširiti po vsej mreži vse do vhodnega nivoja, saj se učijo vsi sloji. Ta postopek v kontektsu nevronskih mrež imenujemo vzvratno razširjanje napake (angl. *backpropagation of error*), temelji pa na posplošenem matematičnem pravilu delta.

Pred razvojem algoritma za vzvratno razširjanje napake so se za iskanje optimalnih uteži uporabljali različni postopki. Po enem izmed njih je mreža naključno spremenila uteži in v primeru, da se je rezultat poslabšal, naslednjič posodobila uteži z nasprotno vrednostjo, zmanjšala velikost spremembe ali pa uporabila kombinacijo obeh transformacij. Drugi pristop je bil iska-

nje optimuma s pomočjo genetskih algoritmov. Učenje je bilo dolgotrajno in neoptimalno, saj smer in velikost spremembe uteži nista bili določeni analitično. Zato je razvoj vzvratnega razširjanja napake eden najpomembnejših mejnikov v razvoju nevronskih mrež.

Idealno bi bila funkcija napake konveksna funkcija, cilj učenja pa poiskati ustrezno kombinacijo uteži, pri kateri bi bila napaka najmanjša. Ker pa funkcija napake ni konveksna, poleg tega pa je tudi neznana kompleksna funkcija, je njen minimum nemogoče najti tako, da njen odvod enačimo z nič. Zato je v vsakem koraku potrebno poiskati lokalni gradient funkcije in uteži prilagoditi v smeri nasprotni gradientu s pomočjo postopka imenovanega spust po gradientu (angl. *gradient descent*).

Ker je v praksi izredno zahtevno analitično izračunati odvod celotne funkcije, ki je sestavljena funkcija večih uteženih vsot, nelinearnih funkcij, normalizacije, in drugih je potrebno postopek spusta po gradientu izvesti postopoma, za vsako operacijo ločeno in odvod v vsaki stopnji izračunati numerično. Matematično verižno pravilo za odvajanje pove, kako posredno odvajati kompozitum funkcij $z(y(x))$ glede na spremenljivko x, kadar funkcije ne znamo odvajati direktno po tej spremenljivki:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y}\frac{\mathrm{d}y}{\mathrm{d}x}. \tag{2.8}$$

Po zgledu verižnega pravila lahko napako razširjamo po mreži v obratni smeri, kar močno poenostavi postopek učenja, saj so vse funkcije v mreži enostavno odvedljive. Vsak nevron na podlagi lokalnega gradienta prilagodi vrednosti uteži w_i po formuli

$$w_i = w_i - \eta\frac{\delta L}{\delta w_i}, \tag{2.9}$$

kjer je η hitrost učenja, $\frac{\delta L}{\delta w_i}$ pa občutljivost funkcije napake $L(\cdot)$ na spremembe uteži w_i. Za hitrejšo in boljšo optimizacijo, se pogosto uporablja postopek *momentum*. Sprememba uteži se izračuna kot razlika hitrosti iz prejšnjega koraka, znižane za faktor μ, ter drugega člena v Izrazu (2.9).

2.1.3 Funkcija napake

Na področju reševanja optimizacijskih problemov funkcija napake (angl. *loss function* ali *cost function*) predstavlja ceno netočnosti pri klasifikaciji [20]. Namen optimizacije je minimizirati funkcijo napake in s tem izboljšati točnost modela. Funkcija napake izmeri razliko med napovedjo modela in resnično oznako učnega primera. Skupna napaka je povprečje napak na posameznih učnih primerih.

Najpogostejša in najpreprostejša funkcija napake je Evklidska napaka, imenovana tudi vsota kvadratov (angl. *sum-of-squares*). Njena matematična formula je

$$L(x) = \frac{1}{2N} \sum_{i=1}^{N} (x_i - t_i)^2, \tag{2.10}$$

kjer x predstavlja vhod v funkcijo napake, v tem primeru napoved modela, t pa pravilno oznako. Konstanta $\frac{1}{2}$ na dejansko obliko funkcije ne vpliva, poenostavi pa njen odvod.

Na zadnjem nivoju nevronske mreže, ki poda končne vrednosti za vsak razred, se namesto običajne aktivacijske funkcije pogosto uporablja funkcija *softmax*. Izhodne vrednosti, ki so pozitivne in katerih vsota je ena, tvorijo verjetnostno porazdelitev. V kombinaciji s funkcijo softmax se pogosto uporablja katera izmed variacij logistične funkcije napake (angl. *logistic loss*).

2.1.4 Regularizacija

Nevronske mreže z velikim številom parametrov lahko zajamejo veliko podrobnosti učne množice, ki niso nujno značilne za nek razred, zato pogosto pride do pretiranega prileganja učnim primerom (angl. *overfitting*). Rezultat je model, ki na učni množici dosega visoko točnost, vendar slabo klasificira nove primere. Možne rešitve problema so povečanje učne množice, manjšanje števila parametrov mreže in predčasno ustavljanje učenja (angl. *early stopping*). Prva rešitev ni vedno možna, saj je težko priti do ustrezno označenih podatkov, druga ne izkoristi polnega učnega potenciala nevronske mreže, predčasno ustavljanje učenja pa zahteva eksperimentalno določitev

točke ustavljanja, ki ni nujno optimalna in zanesljiva [21]. Zato se za preprečevanje pretiranega prileganja pogosto uporablja družina postopkov, imenovana regularizacija (angl. *regularization*).

Pogosta izbira je L_2 regularizacija. Ta funkciji napake doda člen, vsoto kvadratov uteži, ki "kaznuje"visoke uteži. Formula regularizirane funkcije napake $L(x,w)$, ki postane funkcija aktivacij x in uteži w, je

$$L(x,w) = L_0(x) + \frac{\lambda}{2n} \sum_w w^2, \tag{2.11}$$

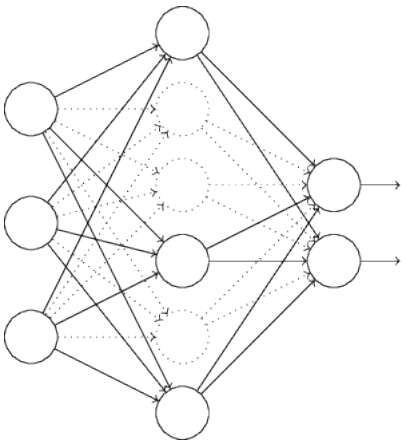
kjer je $L_0(x)$ osnovna funkcija napake, λ pa regularizacijski parameter. Visoka vrednost regularizacijskega parametra pri učenju daje večjo vrednost visokim utežem, nizka pa zmanjša vpliv uteži na celotno vrednost funkcije. Manj pogosto se uporablja L_1 regularizacija, ki namesto kvadratov uteži sešteje njihove absolutne vrednosti.

Pogosto, lahko tudi v kombinaciji z L_2 regularizacijo, se uporablja tudi metoda osipa (angl. *dropout*). Za razliko od prejšnjih, ta direktno ne spreminja funkcije napake. V vsaki iteraciji je naključno izbran del nevronov, ki v tej iteraciji niso aktivni (Slika 2.3). Tako se na določenem paketu slik uči samo del mreže, kar ima podoben učinek kot učenje več mrež na istih podatkih, z različnimi začetnimi utežmi, in povprečenju njihovih izhodov za končno klasifikacijo.

2.2 Konvolucija

Konvolucija je matematična operacija, ki se zelo pogosto uporablja pri procesiranju slik. Glajenje, ostrenje, zmanjševanje šuma ter zaznavanje robov so le nekatere od najpogostejših uporab konvolucije v praksi.

Vsaka točka v izhodu konvolucije (g) je rezultat funkcije istoležne in sosednih točk v vhodni sliki (I). Omenjena funkcija je definirana v konvolucijskem filtru oziroma jedru (h). Konvolucija se v enačbah prikazuje z znakom $*$. Naj bo $\mathbf{I}(i,j)$ intenzitetni nivo slike (I) na koordinatah (i,j), $\mathbf{h}(k,l)$ pa intenzitetni nivo konvolucijskega jedra na koordinatah (k,l). Rezultat konvolucije slike $\mathbf{I}$



Slika 2.3: Pri regularizaciji z osipom je del mreže neaktiven. Slika povzeta po [21].

z jedrom **h** na koordinatah (i,j) je definiran kot

$$g(i,j) = I * h = \sum_{k,l} I(i-k, j-l)h(k,l). \tag{2.12}$$

Ker konvolucijo lahko izračunamo samo, kadar je celoten filter znotraj slike, je rezultat konvolucije slika, ki je za polovico velikosti filtra manjša od vhodne slike. Temu se je mogoče izogniti z dodajanjem ničel ob rob slike.

Konvolucija je komutativna in asociativna operacija. Poleg tega spada med operacije, ki so linearne in neodvisne od zamika (angl. *linear shift-invariant*, v nadaljevanju LSI). To pomeni, da zanjo veljata princip o superpoziciji (2.13) in neodvisnosti od premika (2.14). Slednji pove, da operator deluje enako, ne glede na to kje v sliki se trenutno nahaja. V spodnjih enačbah operator $\circ$ predstavlja LSI operacijo:

$$h * (f_0 + f_1) = h \circ_f 0 + h \circ f_1, \tag{2.13}$$

$$g(i,j) = f(i+k, j+l) \iff (h \circ g)(i,j) = (h \circ f)(i+k, j+l). \tag{2.14}$$

V Tabeli 2.1 so navedeni primeri filtrov, njihov vpliv na sliko pa je prikazan v Sliki 2.4.

Filter zameglitve s povprečenjem	$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$
Filter za zaznavanje robov	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
Filter ostrenja	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Tabela 2.1: Primeri preprostih filtrov.

2.3 Konvolucijske nevronske mreže

Pri reševanju problemov na področju procesiranja slik, besedila in govora umetne nevronske mreže niso dovolj zmogljive, saj se naučijo strukture slik v učni množici, ne znajo pa razpoznavati objektov ne glede na to kje v sliki se nahajajo. ANN je zmožna, na primer, prepoznati sliko centrirane številke 8, če se je učila na takšnih podatkih, ne razpozna pa številke 8, ki se nahaja v zgornjem desnem kotu, če takšne slike ni v učni množici. Po zgledu delovanja živalske optične skorje (angl. *visual cortex*) so se kot nadgradnja ANN razvile konvolucijske nevronske mreže, ki celotno sliko "razkosajo"na prekrivajoča se sprejemna polja (angl. *receptive field*). Sprejemno polje nevronov na najnižjem nivoju, ki sledi vhodnemu nivoju, je zelo majhno in pokriva samo nekaj sosednih slikovnih točk, medtem ko sprejemno polje vsakega naslednjega nivoja pokriva večji del vhodne slike.

Vhodni podatki pri problemih računalniškega vida so ponavadi oblike $W \times H \times D$, kjer sta W in H širina in višina (v nadaljevanju prostorski dimenziji), D pa število barvnih kanalov, ki je najpogosteje enako 1 pri črnobelih

Slika 2.4: Originalna slika (levo zgoraj), konvolirana z različnimi filtri.

in sivinskih slikah ter 3 pri barvnih slikah v RGB shemi.

Arhitekture konvolucijskih nevronskih mrež se razlikujejo od problema do problema. Optimalna arhitektura se določa s pomočjo validacijske množice. Pri sestavljanju CNN se najpogosteje uporabljajo spodaj opisane vrste nivo-

jev.

2.3.1 Nivoji

V tem poglavju so opisani najpomembnejši gradniki konvolucijskih nevronskih mrež: konvolucijski nivo, nivo ReLU, nivo združevanja, polno povezani nivo. Temeljni nivo konvolucijske nevronske mreže je **konvolucijski nivo**. Parametre konvolucijskega nivoja predstavlja množica konvolucijskih jeder oziroma filtrov, ki imajo majhno sprejemno polje (širino in višino) in se raztezajo čez celotno globino vhodne slike oziroma matrike aktivacij iz prejšnjega nivoja. Pri prehodu naprej (angl. *forward pass*) konvolucija vsakega jedra z vhodom vrne dvodimenzionalno polje, zato je globina izhoda iz konvolucijskega nivoja enaka številu filtrov, medtem ko sta širina in višina odvisni od velikosti in prekrivanja sprejemnih polj.

Enako kot uteži pri ANN, so filtri pri CNN naključno inicializirani in predstavljajo učeči se del mreže. V fazi učenja se naučijo razpoznavati različne značilnosti v sliki, na primer različno orientirane robove in področja slikovnih točk iste barve (angl. *blob*). Ker pri večini problemov v računalniškem vidu struktura slike ni vnaprej določena in se lahko določene značilke pojavljajo v kateremkoli delu slike, si vsi nevroni na istem nivoju delijo filtre - uteži in pristranskost (angl. *bias*), kar drastično zmanjša število parametrov, ki se jih mora mreža naučiti.

Konvolucijskemu nivoju je potrebno določiti tri hiperparametre, ki poleg velikosti filtra definirajo velikost izhoda. Prvi je globina izhoda, ki je enaka številu filtrov v nivoju. Drugi hiperparameter je velikost koraka (angl. *stride*), s katerim premikamo filter vzdolž vhoda. Če je ta enaka 1, se konvolucija računa za vsako slikovno točko, pri vrednosti 2 pa na vsaki drugi slikovni točki. Vrednosti večje od 2 se v praksi ne uporabljajo pogosto. Večja velikost koraka pomeni manjši prostorski dimenziji izhoda. Tretji hiperparameter je obrobljanje z ničlo (angl. *zero-padding*), ki definira število ničelnih robov dodanih vhodu v nivo. S pomočjo tega hiperparametra lahko kontroliramo velikost izhoda. S pravilno nastavitvijo je mogoče ohraniti prostorske

dimenzije vhoda, saj je rezultat konvolucije sicer vedno manjši od njenega vhoda.

Število nevronov v nivoju je enako $(W-K+2P)/S+1$, kjer je W velikost vhoda, K velikost konvolucijskega filtra, P število ničelnih robov in S velikost koraka. Če rezultat ni celo število, je potrebno vrednosti hiperparametrov prilagoditi. Pri vzvratnem prehodu (angl. *backward pass*) se uporablja isti filter, le da sta njegova širina in višina zamenjani.

Aktivacijska funkcija v konvolucijskih nevronskih mrežah je ponavadi ReLU (prikazana v Izrazu (2.6)), saj je preprosta in hitra za izračun, zato je nivo, ki praviloma sledi konvolucijskemu nivoju imenovan **ReLU nivo**. Izhod ReLU nivoja je enakih dimenzij kot izhod iz konvolucijskega nivoja pred njim, le da so vrednosti nelinearno transformirane.

Nivo združevanja (angl. *pooling layer*) zmanjšuje prostorske dimenzije slike, s čimer reducira število parametrov v višjih nivojih in obenem preprečuje pretirano prilagajanje učnim primerom. Najpogosteje se uporabljajo filtri velikosti 2×2 s korakom 2. Takšni filtri izvedejo preprosto operacijo na neprekrivajočih se oknih velikosti 2×2. Ponavadi se uporablja operacija $\max(\cdot)$, ki izbere najvišjo vrednost v oknu, v preteklosti pa je bila pogosta izbira tudi $\text{avg}(\cdot)$, ki izračuna povprečje vrednosti v oknu.

Ker opreacije združevanja uporabljajo vnaprej definirane filtre, se ta nivo ne uči, zato ne prispeva k skupnemu številu parametrov, ki se jih mreža uči. V praksi si med prehodom naprej mreža zapomni indeks najvišje vrednosti v oknu, saj predstavlja pomembno informacijo pri vzvratnem razširjanju napake.

Vsi nevroni v **polno povezanem nivoju** so povezani z vsemi aktivacijami iz prejšnjega nivoja. Ponavadi se uporabljajo na vrhu nevronske mreže in so namenjeni izračunu končnih verjetnosti za vsak razred. Polno povezani nivo je pogosto implementiran kot konvolucijski nivo, katerega filtri so enake velikosti kot vhodne aktivacije.

Najpogosteje je CNN sestavljena iz več zaporednih parov konvolucijskega (CONV) in ReLU nivoja, ki jim sledi nivo združevanja (POOL). To zaporedje

je nato lahko večkrat ponovljeno, sledi jim nekaj polno povezanih nivojev (FC), skupaj s pripadajočimi ReLU nivoji in na koncu še en polno povezani nivo. Izraz

$$INPUT \rightarrow [[CONV \rightarrow RELU] * N \rightarrow POOL?] * M \\ \rightarrow [FC \rightarrow RELU] * K \rightarrow FC \quad (2.15)$$

najenostavneje povzame opisano zaporedje. V izrazu N, M in K predstavljajo poljubna naravna števila, znak "?" pa opcijski nivo.

2.3.2 Priprava podatkov

Pomemben korak za hitrejše in učinkovitejše učenje nevronske mreže je pravilna predpriprava podatkov in razdelitev na množice. V skladu z metodami umetne inteligence je potrebno podatke razdeliti vsaj na učno in testno množico, v primeru, ko je potrebno iskanje ustreznih parametrov pa še validacijsko množico. Mreža se uči na podatkih v učni množici, nato pa končni model preizkusimo na testni množici, ki mora biti sestavljena iz novih primerov, saj lahko le tako dobimo vtis o točnosti, ki jo bo model dosegal na nevidenih podatkih. Kadar določamo parametre učenja, kot sta hitrost učenja in momentum, je potrebno uporabiti še validacijsko množico, na kateri preverjamo točnost pri različnih vrednostih določenega parametra, testno množico pa uporabimo šele za končni izračun točnosti, saj pri iskanju optimalnih vrednosti parametrov lahko pride do pretiranega prilagajanja (angl. *overfitting*) validacijske množice. Zelo pomembno je, da so množice čimbolj uravnotežene z vidika porazdelitve razredov, ter da so deleži razredov v različnih množicah podobni.

Metode za predpripravo podatkov so številne, njihova uporaba pa je močno odvisna od lastnosti podatkov. Pri delu s konvolucijskimi nevronskimi mrežami, sta najpomembnejša odštevanje srednje vrednosti in normalizacija.

Pri odštevanju srednje vrednosti se vsaki značilki (angl. *feature*) odšteje srednja vrednost te značilke v vseh učnih primerih, rezultat tega pa je centriranje podatkov okrog koordinatnega izhodišča. Pri slikah za učenje CNN

se ponavadi vsem slikovnim točkam odšteje ista srednja vrednost vseh slik, oziroma ločeno glede na barvni kanal.

Normalizacija je postopek, s katerim zagotovimo, da so vrednosti vseh značilk na približno istem intervalu. S tem preprečimo, da bi značilke z večjo zalogo vrednosti dobile višje uteži kot značilke z vrednostmi na manjšem intervalu in bile s tem upoštevane kot pomembnejše. Normalizacijo lahko izvedemo tako, da vrednosti preslikamo na interval -1 do 1 ali vrednosti najprej centriramo okrog ničle in nato delimo s standardnim odklonom po vsaki dimenziji. Ker pri delu s slikami lahko predpostavimo, da so vse vrednosti na intervalu med 0 in 255, normalizacija ni nujno potreben korak predpriprave podatkov za učenje CNN.

Parametre predpriprave (srednjo vrednost in standardni odklon) je potrebno izračunati na učni množici po izvedeni delitvi podatkov na množice, nato pa obdelati validacijsko in testno množico s parametri učne množice.

2.3.3 Naučeni modeli

V praksi se konvolucijske nevronske mreže zelo redko učijo od začetka, ko so uteži naključno inicializirane. Razlog za to je, da ponavadi ne obstaja dovolj velika označena podatkovna zbirka za specifičen problem, izdelava takšne zbirke pa zahteva zelo veliko časa. Zato se mreža pogosto najprej uči na splošnejši podatkovni zbirki, kot je ImageNet [9], ki vsebuje več kot miljon označenih slik iz 1000 kategorij, nato pa se model prilagodi specifičnemu problemu.

Vnaprej naučenemu modelu nato lahko odstranimo najvišji polno povezani nivo in ga nadomestimo z naključno inicializiranim polno povezanim nivojem ali pa namesto njega preprosto uporabimo metodo podpornih vektorjev ali podoben linearni klasifikator. Spodnji nivoji mreže namreč iščejo bolj splošne značilke kot so robovi, medtem ko zgornji nivoji zaznavajo bolj specifične značilnosti slike. Druga možnost je, da doučimo (angl. *finetune*) del mreže z manjšo hitrostjo učenja. Število učečih se nivojev je odvisno od števila podatkov in njihove podobnosti splošni podatkovni zbirki.

Poglavje 3

Razpoznavanje prostorov s segmentacijo

Mreža, naučena za namen te diplomske naloge, temelji na članku avtorja L.-C. Chen et al., 2015 [16]. Implementacija mreže, predstavljene v tem članku, temelji na ogrodju Caffe za učenje globokih nevronskih mrež. Prvi del tega poglavja je namenjen pregledu ogrodja, drugi del opisuje mrežo opisano v članku, na koncu pa je predstavljen naš pristop.

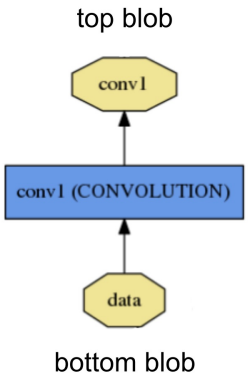
3.1 Ogrodje Caffe

Ogrodje Caffe [15] ločeno implementira vsak nivo mreže in jih nato poveže prek vhodnih in izhodnih podatkov. Za shranjevanje in posredovanje podatkov med nivoji je v ogrodju Caffe definirana struktura imenovana blob. Z uporabo teh struktur so poenoteno zapisane skupine slik v obdelavi, parametri mreže in odvodi pri optimizaciji, kar omogoča preprostejšo interakcijo med različnimi tipi podatkov ter efektivno sinhronizacijo med procesiranjem na centralni in grafični procesni enoti.

Matematično je struktura blob definirana kot N-dimenzionalno polje. Pri delu s slikami je ta najpogosteje 4-dimenzionalna, uporablja pa se lahko tudi za drugačne vrste podatkov. Za vhodne slike je dimenzija enaka $N \times K \times H \times$

W, kjer je N število slik v paketu (angl. *batch*), K število barvnih kanalov, H in W pa višina in širina slike.

Vsak nivo potrebuje definirano vhodno (polje *bottom*) in izhodno (polje *top*) povezavo, ki omogoča komunikacijo z ostalimi nivoji v mreži, kot prikazuje Slika 3.1. Vsak nivo omogoča prehoda naprej in nazaj čez mrežo. Med prehodom naprej nivo prejme matriko aktivacij prejšnjega nivoja, izvede svojo operacijo nad njimi in jo posreduje kot izhod sledečim nivojem. Med vzvratnim prehodom sprejme kot vhod od višjeležečega nivoja gradient glede na svoj izhod, izračuna gradiente glede na svoje parametre ter vhode in jih posreduje nižjeležečim nivojem. Preprosta implementacija prehodov v obe smeri je posledica lepih lastnosti postopka vzvratnega razširjanja napake, ki je opisan v Poglavju 2.1.2.



Slika 3.1: Primer vhodnega (*data*) in izhodnega (*conv1*) bloba za konvolucijski nivo *conv1*.

Ogrodje Caffe omogoča definicijo novih tipov nivojev, saj vsakemu nivoju priprada ločena datoteka napisana v programskem jeziku C++. Mreža je implementirana kot usmerjen nepovezan graf, ki se tipično začne s podatkovnim nivojem in konča z nivojem, ki izračunava izgubo (angl. *loss layer*). Definicija mreže je ogrodju Caffe podana preko datoteke tipa prototxt. Primer definicije nivoja združevanja, s filtrom velikosti 2×2, korakom 2 in širino robu ničel 1:

```
layers {
  bottom: "conv1_2"
  top: "pool1"
  name: "pool1"
  type: POOLING
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
    pad: 1
  }
}
```

V povezavi z ogrodjem Caffe je na internetu knjižnica modelov, imenovana Model ZOO, kamor uporabniki lahko prosto nalagajo naučene modele. Tako lahko uporabniki brez zadostne računske moči uporabijo obstoječ model in ga prilagodijo svojim potrebam. Slabost takšnega pristopa je, da je struktura mreže vnaprej določena in ne omogoča veliko predelav predvsem v nižjih (konvolucijskih) nivojih.

3.2 Mreža Deeplab

Kot osnova za učenje segmentacije prostorov služi mreža Deeplab [16], ki je mreža za semantično segmentacijo. Temelji na 16-nivojski klasifikacijski mreži imenovani VGG-16 [22], naučeni na zbirki Imagenet [9]. Vnaprej naučen model je prilagojen na zbirki Pascal VOC 2012, ki ima 21 kategorij.

Mreža je sestavljena iz petih zaporedij konvolucijskih nivojev in nivoja podvzorčenja. Prvi dve zaporedji sestavljata vsako po dva konvolucijska nivoja, zadnje tri pa po trije. Sledijo jim trije polno povezani nivoji, ki so implementirani z uporabo konvolucijskih nivojev. Izhod zadnjega polno povezanega nivoja je vhod v klasifikator Softmax, nad njegovim izhodom pa je izračunana multinomska logistična funkcija napake. Pri testiranju je

resolucija izhodnih klasifikacij interpolirana s faktorjem 8.

Za izboljšavo točnosti segmentacije so avtorji članka na rezultatih segmentacije mreže uporabili polno povezana pogojna slučajna polja (angl. *conditional random fields*, CRF) [25]. CRF delujejo na podlagi verjetnosti, da ima točka x_i oznako l_i glede na okoliške točke in njihove oznake. Pri polno povezanih CRF je okolica kar celotna slika, saj namen ni glajenje segmentacij, ampak rekonstrukcija detajlov lokalne strukture kot prikazuje Slika 3.2.

Slika 3.2: Primera delovanja mreže Deeplab. Vhodni sliki sledi segmentacija iz mreže, nato pa izboljšana lokalizacija objekta z uporabo CRF. Sliki povzeti po [16].

Model je evaluiran na zbirki PASCAL VOC in dosega rezultate, primerljive z drugimi sodobnimi metodami na področju segmentacije. Povprečna vrednost mere preseka nad unijo (angl. *mean intersection over union*, mIOU), ki jo dosega osnovna mreža nadgrajena s CRF je 66,4 odstotkov. Mera IOU se pogosto uporablja za ovrednotenje segmentacijskih problemov, saj zajame razmerje med številom pravilno in nepravilno klasificiranih točk. Vrednost preseka sama po sebi ni merodajen podatek, saj lahko s klasifikacijo celotne slike v pravilni razred dobimo visoko vrednost in s tem ignoriramo napačno klasificirane pozitivne vrednosti (angl. *false positive*).

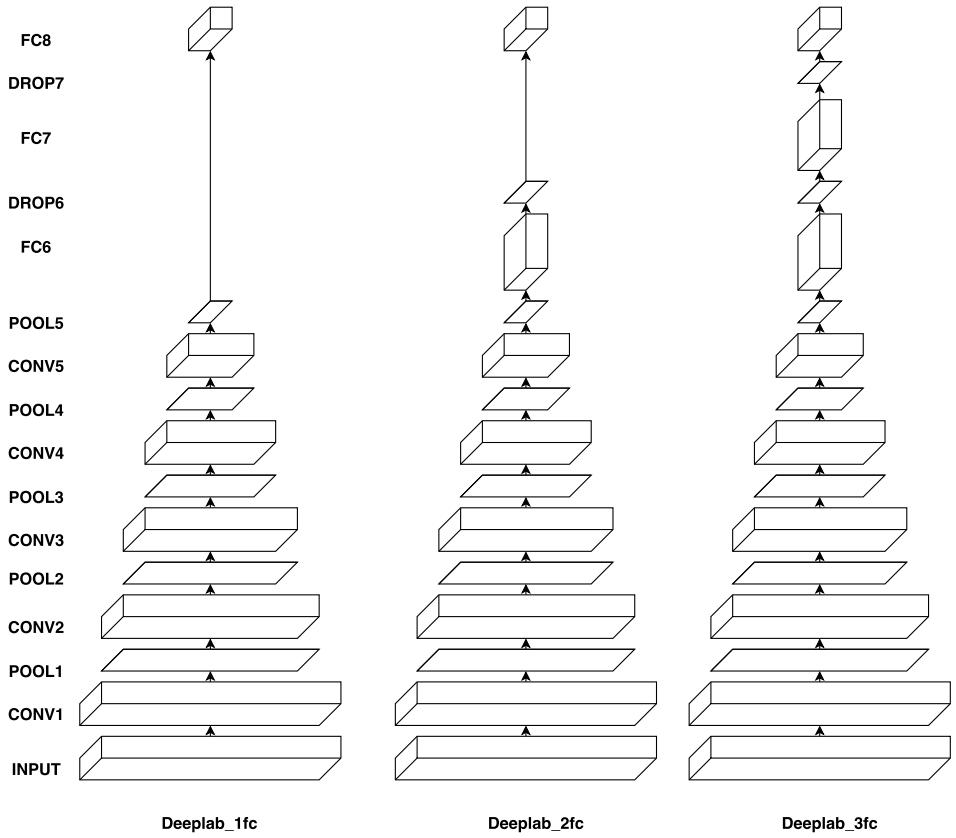
3.3 Razpoznavanje s segmentacijo

V našem delu smo se razpoznavanja prostorov lotili z uporabo semantične segmentacije, ki razred določa na nivoju slikovnih točk, in s tem naslovili slabost obstoječih metod v primeru, da se na sliki pojavlja več prostorov. V

ta namen smo uporabili model konvolucijske nevronske mreže. Ker učenje celotne nevronske mreže zahteva veliko podatkov in računske moči, smo kot osnovo v skladu s postopki, opisanimi v Poglavju 2.3.3 uporabili arhitekturo segmentacijske mreže Deeplab [16], opisane v prejšnjem poglavju. Uporabili smo nespremenjene konvolucijske nivoje, ki se naučijo zaznavati splošne značilnosti, na primer robove, ki so neodvisne od specifične naloge. Učenje teh nivojev bi lahko privedlo do prevelikega prilagajanja učnim podatkom.

Z namenom preverjanja vpliva števila polno povezanih nivojev smo ustvarili tri arhitekture mrež. Mreža Deeplab_3fc ima tri polno povezane nivoje, enako kot mreža Deeplab. Deeplab_2fc ima izhod prvega polno povezanega nivoja (v nadaljevanju*fc6*) povezan s tretjim polno povezanim nivojem (v nadaljevanju *fc8*), torej preskoči drugi polno povezani nivo (v nadaljevanju *fc7*). Mreža Deeplab_1fc ima samo en polno povezani nivo, njegov vhod pa je izhod iz nivoja združevanja na petem nivoju (*pool5*), ki sledi zadnjemu nivoju konvolucije. Arhitektura opisanih mrež je za večjo preglednost prikazana na Sliki 3.3 Zadnji polno povezani nivo (fc8) v vseh mrežah smo naključno inicializirali in učili od začetka, ostale nivoje pa smo samo modificirali z neko majhno hitrostjo učenja.

Izhod mreže je večdimenzionalno polje, v katerem vsaki slikovni točki pripada vektor z verjetnostmi za posamezne razrede. Za evaluacije metod smo vsako točko uvrstili v najverjetnejši razred tako, da smo dobili enodimenzionalno polje v velikosti vhodne slike, kjer je vrednost vsake točke oznaka istoležne točke v vhodni sliki. Rezultatom testiranja smo nato izračunali točnost določanja razredov na nivoju slikovnih točk tako, da smo preverjali, katere kategorije predstavljajo dovolj velik delež slike, da jih lahko obravnavamo kot detekcije in ne kot šum.



Slika 3.3: Arhitekture predlaganih mrež. Zaradi ujemanja z besedilom so imena nivojev v angleščini, kjer je DATA vhodni nivo, CONV konvolucijski nivo, POOL nivo združevanja, FC polno povezani nivo in DROP nivo osipa.

Poglavje 4

Rezultati

V tem poglavju je najprej opisana podatkovna zbirka, ki je bila uporabljena za učenje in testiranje modelov. Sledi predstavitev parametrov učenja in analiza funkcije napake. Tretji del poglavja je namenjen predstavitvi postopkov vrednotenja rezultatov in uporabljenih mer, sledi pa mu kvantitativna analiza. Poglavje se zaključi s kvalitativno analizo.

4.1 Učenje

4.1.1 Podatki

Kot osnovo za podatkovno zbirko uporabljeno v tej diplomski nalogi smo izbrali zbirko , ki so jo na univerzi MIT razvili za potrebe članka iz leta 2009 [1]. Sestavljena je iz približno 15000 slik, ki pripadajo 67 kategorijam prostorov. Vsaki kategoriji pripada najmanj 100 slik. Zbirka je namenjena klasifikaciji prostorov, vsebuje pa tudi anotacije posameznih predmetov v slikah.

V diplomski nalogi je uporabljenih samo 8 najpogostejših kategorij prostorov, ki se nahajajo v stanovanju. Te so kopalnica, spalnica, otroška soba, hodnik, garderoba, kuhinja, dnevna soba in jedilnica. Razlike med številom primerov v posamezni kategoriji so velike, zato smo najmanj pogostim prostorom dodali še nekaj dodatnih primerov.

Slika 4.1: Prikaz barvnega kodiranja, uporabljenega za bolj pregledno vizualizacijo rezultatov.

Ker je bil namen diplomske naloge naučiti model za segmentacijo različnih vrst prostorov na slikah, je bilo potrebno oznake spremeniti iz ene oznake na sliko v eno oznako na slikovno točko. Tako vsakemu primeru vhodne slike pripada pravilna segmentacija dimenzij vhodne slike, shranjena v formatu png kot sivinska slika. Vrednosti anotacije so od 0, ki ustreza ozadju, do 8. Vrednosti od 1 do 8 ustrezajo imenov uporabljenih prostorov, v angleščini razvrščenih po abecednem redu, torej od kopalnice do dnevne sobe. V zbirki se je že pojavljalo nekaj slik, na katerih je prikazan več kot en prostor, nekatere pa smo še dodali. Za pohitritev učenja in večjo konsistentnost smo vsem slikam velikost spremenili na 256×256, kar je bila najmanjša resolucija, ki se je pojavljala v bazi.

Za lažjo vizualizacijo smo v programskem okolju Matlab definirali barvno tabelo (angl. *colormap*), ki vsaki vrednosti med 0 in 8 pripiše barvo v prostoru RGB. Omenjeno barvno kodiranje je prikazano na Sliki 4.1.

Za povečanje relativno majhne zbirke smo iz obstoječih slik generirali nove primere. Iz vsake slike, ki ima v originalni zbirki resolucijo večjo od 300×300 slikovnih točko, smo naključno izrezali odsek velikosti 256×256 in jo zarotirali za naključen kot med $1°$ in $40°$ v levo in desno ter zmanjšali na

velikost 256×256. Pri slikah z več prostori rotacija ni bila izvedena, saj bi interpolacija pri rotiranju v matlabu pokvarila pravilno segmentacijo slike.

Zbirko smo razdelili na učno, validacijsko in testno množico tako, da je razporeditev razredov približno enaka v vseh treh množicah. Vsega skupaj vsebuje 10415 primerov, od tega jih 2080 pripada validacijski in 2084 testni množici.

4.1.2 Učenje mrež

Pri učenju mreže je uporabljena kombinacija metod, opisanih v Poglavju 2.3.3. Nivoja *fc6* in *fc7* se učita z nizko hitrostjo učenja reda 10^{-3}. Nivo *fc8* je nadomeščen z naključno inicializiranim nivojem. Ta nivo se uči s hitrostjo učenja reda 10^{-2}, saj se uči od začetka. Hitrost učenja se po korakih zmanjšuje za faktor $\gamma = 0{,}1$. Pri učenju je uporabljen momentum μ z vrednostjo 0,9. Vrednosti omenjenih hiperparametrov niso eksperimentalno določene, ampak so uporabljene grobe ocene predlagane v dokumentaciji ogrodja Caffe [15].

Za preizkus vpliva števila iteracij na točnost segmentacije smo na vsakih 20000 iteracij naredili posnetek modela, ki ga je pozneje enostavno uporabiti kot model za testiranje.

4.2 Metode vrednotenja

Za primerjavo z našimi metodami, smo uporabili trenutno najuspešnejšo metodo za razpoznavanje prostorov PlacesNet [7], ki je naučena na zbirki Places205 [7], ki vsebuje 205 kategorij zunanjih in notranjih prostorov. Iz nivoja softmax smo pridobili verjetnosti posameznih razredov. Nekatere kategorije iz naše zbirke so v zbirki Places205 razdeljene na več kategorij, na primer *kitchen* in *kitchenette*, zato smo vse sorodne kategorije preslikali v ustrezno kategorijo izmed naših osmih tako, da smo njihove verjetnosti sešteli in nato normalizirali, da je njihova vsota 1. Verjetnosti prostorov, ki jih nismo mogli uvrstiti v nobeno izmed kategorij, nismo upoštevali pri primerjavi z našo

mrežo in jim avtomatsko pripisali vrednost nič.

4.2.1 Vrednotenje točnosti lokalizacije

Za vrednotenje točnosti segmentacije smo uporabili mero, ki temelji na Jaccardovem koeficientu podobnosti. Ta se v angleščini imenuje *mean intersection-over-union* (mIOU) in se pogosto uporablja za vrednotenje natančnosti semantične segmentacije [23],[24]. Definirana je kot vsota količnika med presekom in unijo pravilne segmentacije in predikcije mreže preko vseh razredov, deljena s številom razredov. Naj bo C_{ij} število slikovnih točk s pravilno oznako i, ki so klasificirani v razred j. Z $G_i = \sum_{j=1}^{L} C_{ij}$ označimo število vseh točk, ki pripadajo i-temu razredu, s $P_j = \sum i = 1^L C_{ij}$ pa število vseh točk, ki jih klasifikator uvrsti v j-ti razred. Formula mere mIOU je

$$\text{mIOU} = \frac{1}{L} \sum_{i=1}^{L} \frac{C_{ii}}{G_i + Pj - C_{ii}}. \tag{4.1}$$

Ker nivo Softmax v mreži PlacesNet vrne samo verjetnosti posameznih razredov, smo najbolj verjetno oznako razširili na velikost slike in tako dobili ustrezno segmentacijo za primerjavo z našimi modeli.

4.2.2 Vrednotenje točnosti klasifikacije

Druga mera ocenjuje kako učinkovita je mreža pri določanju prostorov, ki se pojavijo v sliki, neodvisno od uspešnosti njihove lokalizacije. Pri rezultatih segmentacijskih mrež smo izračunali deleže slikovnih točk po prostorih, pri rezultatih PlacesNet pa verjetnosti pojavljanja vsakega prostora. Kot prostore, ki se pojavijo v sliki smo sprejeli vse, ki imajo vrednost višjo od določenega deleža maksimuma, imenovanega meja. Za mejo smo uporabili različne vrednosti.

Množico sprejetih prostorov za vsako sliko smo primerjali z množico prostorov v pravilni segmentaciji. Preko vseh slik smo prešteli vse pravilno in nepravilno prepoznane prostore in ustvarili matriko zamenjav. Nato smo izračunali preciznost, priklic in mero F. Naj bo TP število pravilno označenih

primerov (angl. *true positive*), FP število primerov, ki so detektirani v sliki (angl. *false positive*), FN pa število tistih, ki se pojavijo v sliki, vendar niso detektirani (angl. *false negative*). Preciznost je izražena kot

$$\mathrm{Pr} = \frac{TP}{TP + FP}, \tag{4.2}$$

priklic kot

$$\mathrm{Rec} = \frac{TP}{TP + FN}, \tag{4.3}$$

mera F pa predstavlja zvezo med preciznostjo in priklicom:

$$\mathrm{F} = 2 \cdot \frac{\mathrm{Pr} \cdot \mathrm{Rec}}{\mathrm{Pr} + \mathrm{Rec}}. \tag{4.4}$$

Vse vrednosti so na intervalu med 0 in 1, kjer 1 pomeni najboljši rezultat, 0 pa najslabši.

4.3 Kvantitativna analiza

4.3.1 Točnost lokalizacije

Mreže z različnim številom polno povezanih nivojev in PlacesNet smo med seboj primerjali z uporabo mere mIOU, opisane v prejšnjem poglavju. Rezultati so prikazani v Tabeli 4.1. Po pričakovanju mreža Deeplab_1fc dosega znatno manjšo točnost (0,750) od preostalih dveh različic mreže Deeplab (Deeplab_2fc dosega 0,845, Deeplab_3fc pa 0,844), vendar še vedno veliko boljšo točnost od mreže PlacesNet (0,480), ki presenetljivo ne dosega niti 50 odstotne točnosti. Glavni razlog za to je dejstvo, da klasificira celotno sliko v en prostor, kot je razvidno iz prve vrste slik na Sliki 4.2. Poleg tega model včasih napačno klasificira tudi slike, na katerih se pojavi samo en prostor, saj je posledično napačna klasifikacija vseh slikovnih točk v sliki, kar ima velik vpliv na vrednost mere mIOU. Sistem odpove predvsem pri slikah, ki so rotirane, kot kažeta druga in tretja vrstica Slike 4.2. Mreži Deeplab_2fc in Deeplab_3fc dosegata približno enako točnost, iz česar bi lahko sklepali, da dodatni povezani nivo nima pretirano velikega vpliva na uspešnost naučenega

Mreža	mIOU
Deeplab_1fc	0,750
Deeplab_2fc	0,845
Deeplab_3fc	0,844
PlacesNet	0,480

Tabela 4.1: Vrednost mere mIOU.

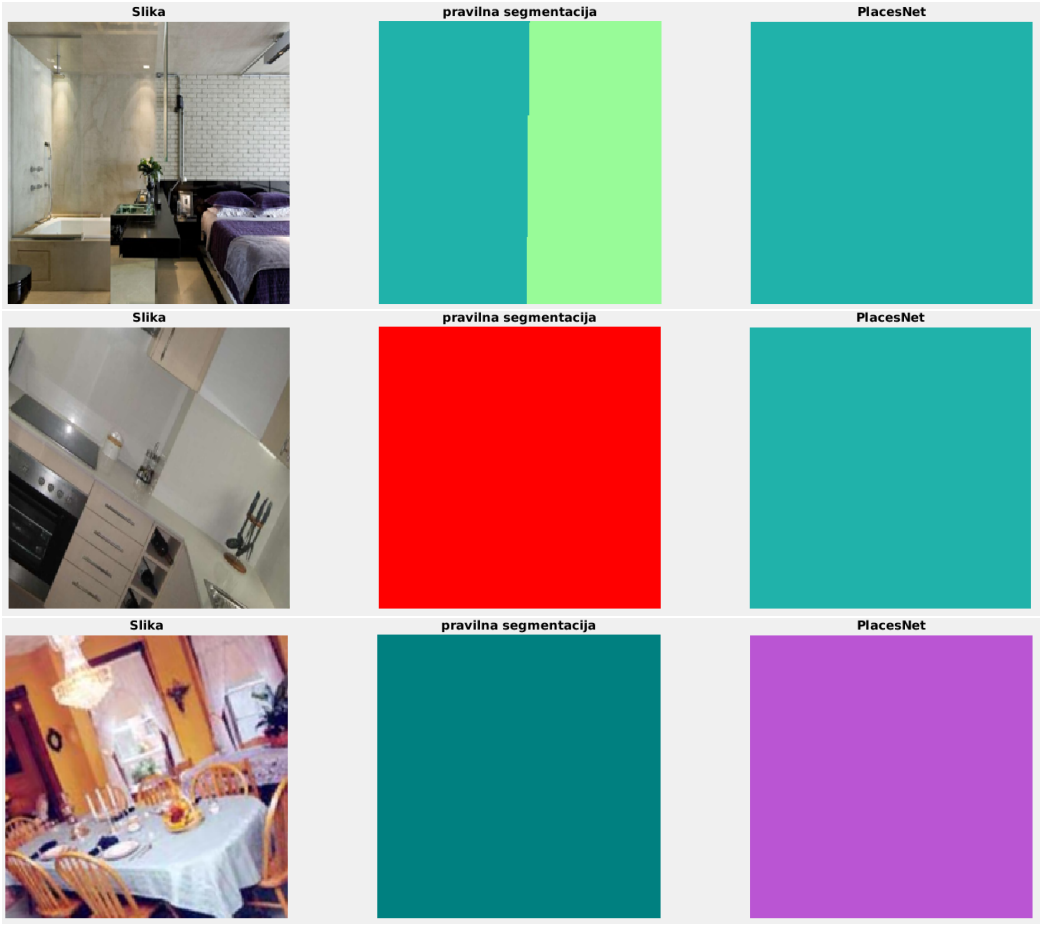
modela. Kljub temu to ni nujno pravilen zaključek, saj bi bil rezultat morda drugačen, če bi mrežo učili z drugačnimi kombinacijami hiperparametrov oziroma več iteracijami.

4.3.2 Točnost razpoznavanja

Izkaže se, da se v odvisnosti od meje za sprejetje določenega razreda, kot je opisano v poglavju o merah za vrednotenje, preciznost rahlo pada pri vseh mrežah do neke vrednosti meje $t = 0{,}96$, nato pa strmo pade (levi graf na Sliki 4.3). Takšna vrednost meje pomeni, da sprejmemo samo razrede, ki jim v sliki priprada več kot 96% površine najštevilčnejšega razreda. To velja za slike, kjer vsaj dva prostora predstavljata skoraj enak delež slike. Ker so slike s takšno porazdelitvijo prostorov redke, večinoma sprejmemo samo najštevilčnejši razred in je verjetnost, da smo izbrali pravi razred, manjša. Z višanjem meje se zmanjšuje število primerov, ki se pojavijo v pravilni segmentaciji, saj postopek izloči prostore, ki zavzemajo manjši delež slike, s tem pa narašča preciznost (desni graf na Sliki 4.3). Mera F je pri vseh vrednostih meje približno konstantna. V Tabeli 4.2 so prikazane vrednosti preciznosti, priklica in mere F pri vrednosti meje $t = 0{,}8$.

4.4 Kvalitativna analiza

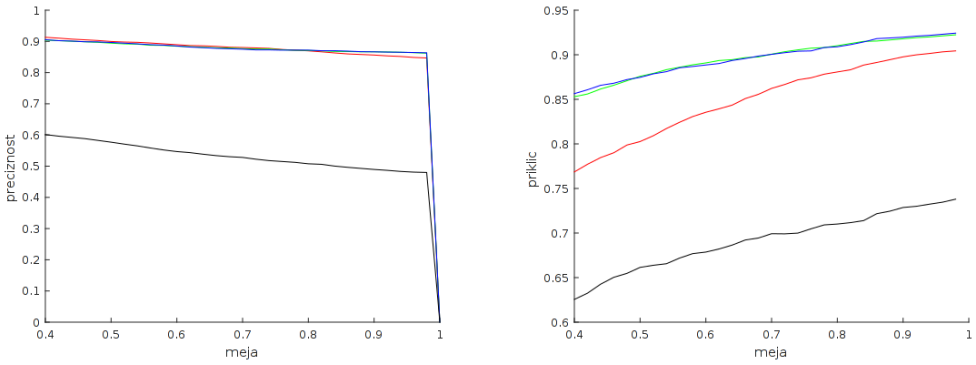
V tem poglavju so rezultati predstavljeni kvalitativno. Rezultati segmentacije so v splošnem sprejemljivi. Vse mreže slike na katerih je prisoten samo en



Slika 4.2: Prvi stolpec je vhodna slika, drugi stolpec je pravilna segmentacija, tretja pa segmentacija pridobljena z mrežo PlacesNet.

Mreža	Preciznost	Priklic	Mera F
deeplab_1fc	0,8697	0,8807	0,8752
deeplab_2fc	0,8710	0,9102	0,8902
deeplab_3fc	0,8719	0,9090	0,8901
placesNet	0,5077	0,7101	0,5921

Tabela 4.2: Preciznost, priklic in mera F pri meji 0,8.



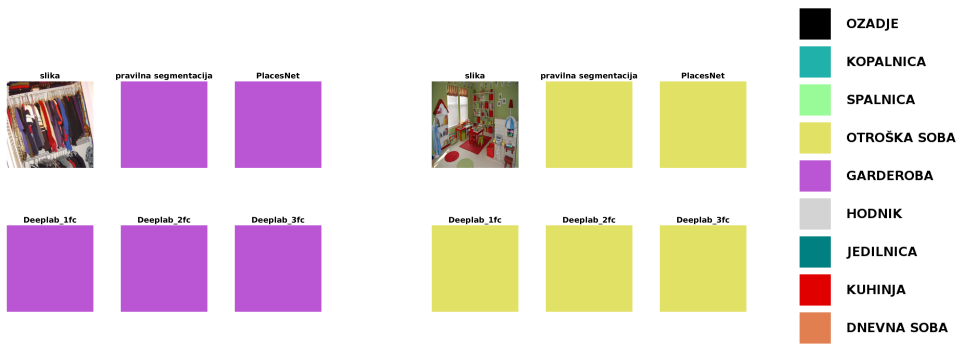
Slika 4.3: Meri preciznost (levo) in priklic (desno) v odvisnosti od meje za sprejetje prostora za vsako mrežo. Mreža Deeplab_1fc je označuje z rdečo krivuljo, Deeplab_2fc z zeleno, Deeplab_3fc z modro in PlacesNet s črno.

prostor pogosto v celoti klasificirajo pravilno, kot prikazuje Slika 4.4. Modeli delujejo tudi pri slikah, ki so zarotirane ali obrezane, če vsebujejo dovolj prepoznavne lastnosti določenega prostora, referenčna metoda PlacesNet pa na takšnih slikah pogosto odpove. Na Sliki 4.5 naši modeli pravilno klasificirajo sliki kuhinje in hodnika, referenčna metoda pa ne.

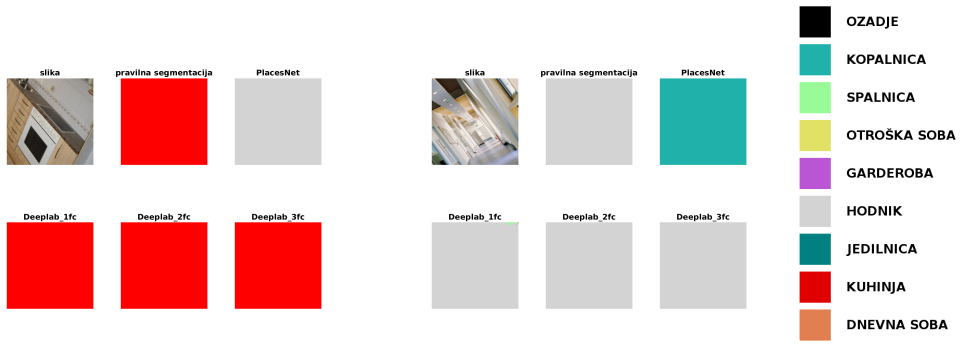
Tudi na slikah z več prostori je segmentacija pogosto kar dobra, v primeru napake pa je ponavadi mogoče rezultat razložiti glede na lastnosti slike. Na Sliki 4.6a z nekoliko šuma mreže pravilno segmentirajo spalnico in garderobo. V zgornjem delu slike pa zaznajo lastnosti hodnika, ki bi jih verjetno brez celotnega konteksta slike napačno klasificiral tudi človek. Primer uspešne segmentacije kuhinje in jedilnice je prikazan na Sliki 4.6b.

Mreži z dvema in tremi polno povezanimi nivoji večinoma delujeta znatno bolje od tiste z enim. Na Sliki 4.7a mreži z dvema in tremi nivoji celotno sliko hodnika uvrstita v pravi razred, segmentacija mreže z enim nivojem pa ob robovih vsebuje veliko šuma. V primeru slike z več prostori na Sliki 4.7b vsi modeli zaznajo prave prostore, vendar sta segmentaciji mrež z dvema in tremi FC nivoji točnejši.

V nekaterih primerih pride do zelo nenatančne segmentacije. V Sliki 4.8a



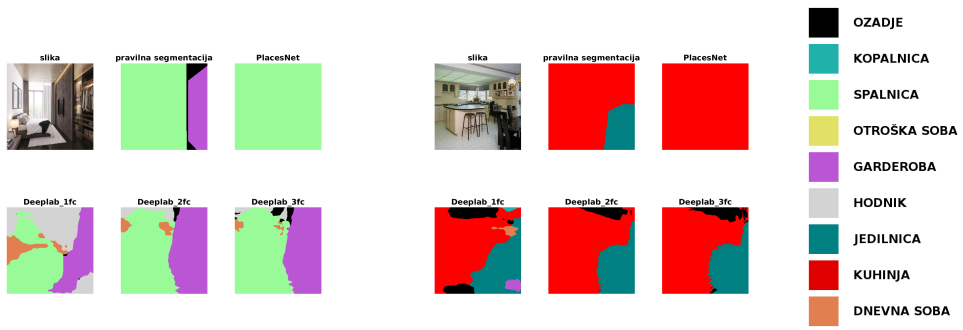
Slika 4.4: Primera dobrih klasifikacij slik z enim samim prostorom. Zgornja vrsta prikazuje sliko, njeno pravilno segmentacijo in rezultat referenčne metode, spodnja pa segmentacijo mrež z enim, dvema in tremi polno povezanimi nivoji, testirane po 100000 iteracijah učenja.



Slika 4.5: Modeli delujejo bolje od referenčne metode v primeru rotacije.

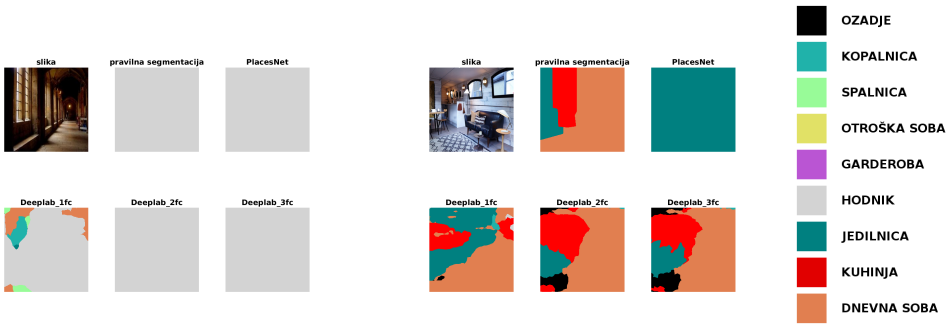
je najverjetneje zaradi barv, tekstur in odbojev svetlobe, ki spominjajo na tiste v kopalnici in spalnici, mreža zelo napačno segmentirala sliko, ki prikazuje kuhinjo in jedilnico. Deeplab_2fc in Deeplab_3fc jedilnico in dnevno sobo na Sliki 4.8b označita točno, zgornji del pa popolnoma napačno razpoznata kot otroško sobo.

Zaradi avtomatskega generiranja umetnih slik so tiste, je iz večjih slik izrezan samo nek zelo majhen detajl. Naši modeli tudi na takšnih slikah večinoma delujejo dobro, kot prikazuje Slika 4.9. Kljub večkratnemu prever-



(a) Segmentacija slike s spalnico in garderobo. Napačna razpoznava hodnika v zgornjem delu slike ni velika in je logično razložljiva.

(b) Segmentacija slike s kuhinjo in dnevno sobo.



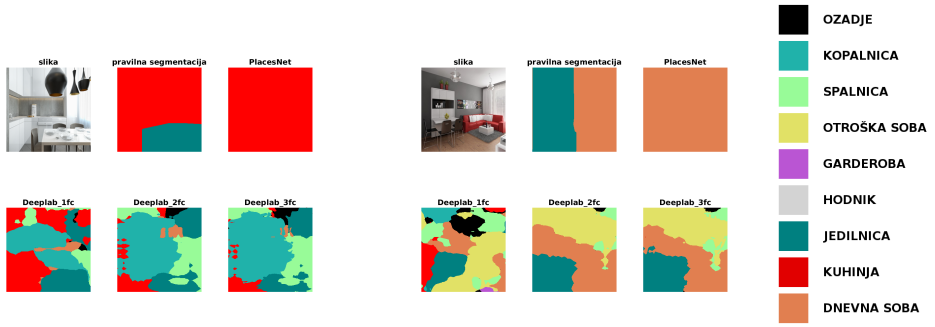
(a) Primer slike z enim samim prostorom.

(b) Primer slike z več prostori.

Slika 4.7: Razlika med rezultatom mreže Deeplab_1fc ter rezultati mrež Deeplab_2fc in Deeplab_3fc je v nekaterih primerih precejšnja, medtem ko slednji delujeta precej podobno.

janju podatkovne zbirke, se v njej še vedno pojavi kakšna, ki je bila v zbirki MIT Indoor uvrščena v en razred, čeprav jih prikazuje več, in ni pravilno segmantirana. V tem primeru včasih modela Deeplab_2fc in Deeplab_3fc podata segmentacijo, ki je "pravilnejša"od segmentacije v zbirki (Slika 4.10).

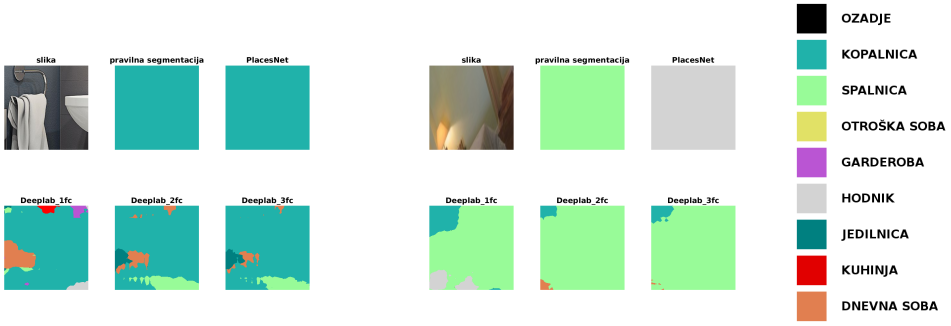
Pregled matrik zamenjav za posamezne modele pokaže, da referenčni mo-



(a) Včasih se kljub človeku očitni klasifikaciji prostora zgodi, da sistem v sliki zazna značilnosti nekega drugega prostora.

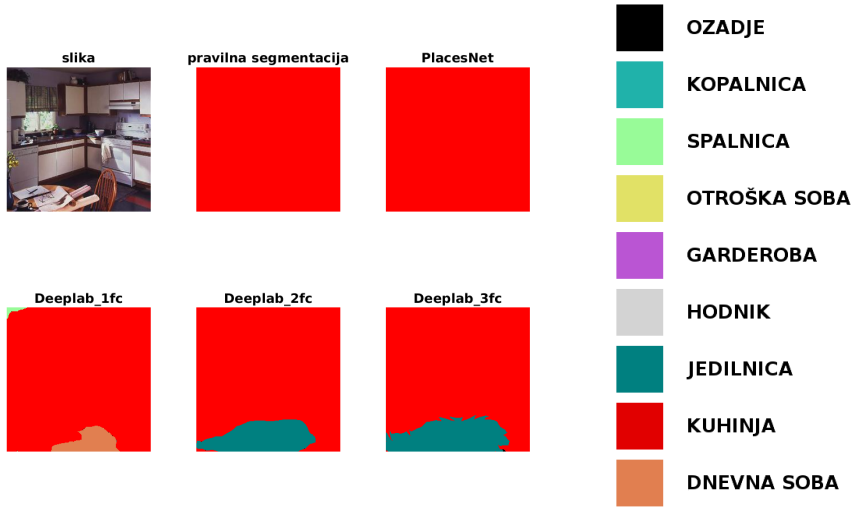
(b) Mreže napačno segmentirajo neznačilne regije. Mreži Deeplab_2fc in Deeplab_3fc dobro zadaneta bistvo v spodnjem delu slike.

Slika 4.8



Slika 4.9: Ker je originalna slika zelo velika, je izrezani detajl velikosti 256 × 256 zelo majhen del osnovne slike, vendar jo mreža še vedno povečini pravilno uvrsti med hodnike.

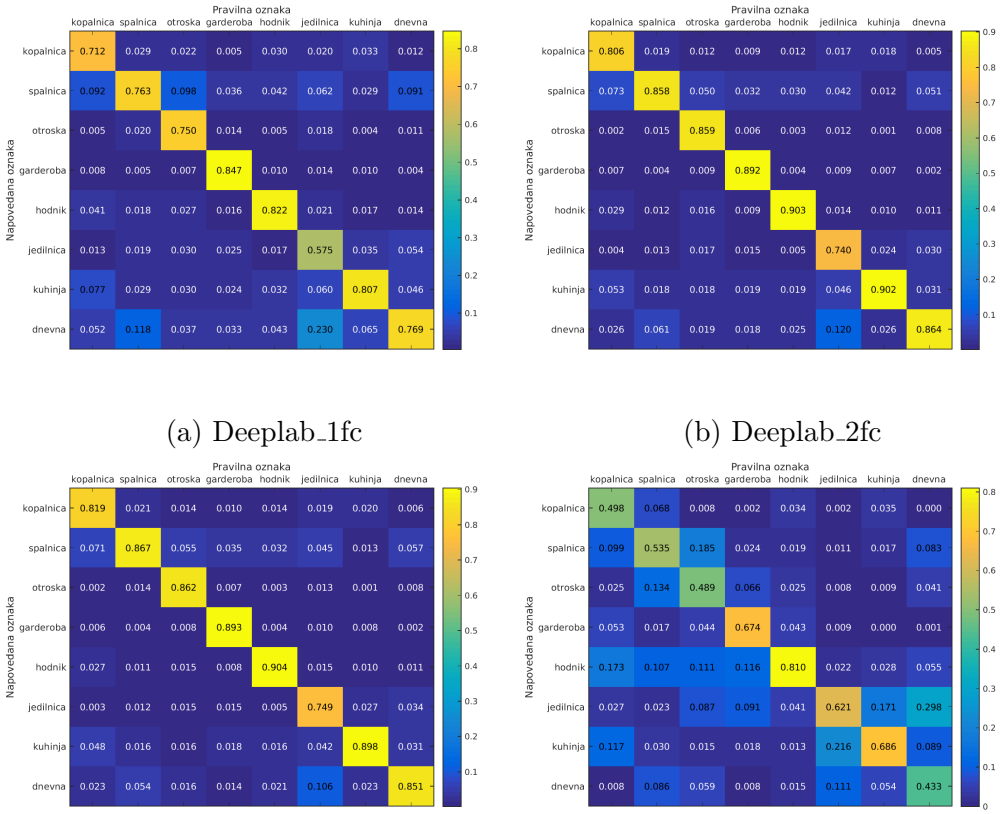
del najpogosteje dnevno sobo za jedilnico, Deeplab_1fc in Deeplab_2fc najpogosteje spalnico označita kot dnevno sobo, Deeplab_3fc pa ravno obratno najpogosteje dnevno sobo označi kot kopalnico. Iz Tabele 4.3 je možno razbrati, da so zelo pogoste zamenjave med prostori, ki so si med seboj zelo podobni in katerih kombinacije se pogosto pojavljajo v istih slikah. Celotne matrike zamenjav za vsako mrežo so prikazane v Sliki 4.11.



Slika 4.10: Slika je v zbirki nepravilno označena, vendar jo dva od treh modelov pravilno segmentirata.

Pravilna oznaka	Deeplab_1fc	Deeplab_2fc	Deeplab_3fc	PlacesNet
Kopalnica	Spalnica	Spalnica	Spalnica	Hodnik
Spalnica	Dnevna soba	Dnevna soba	Dnevna soba	Otroška soba
Otroška soba	Spalnica	Spalnica	Spalnica	Spalnica
Garderoba	Spalnica	Spalnica	Spalnica	Hodnik
Hodnik	Dnevna soba	Spalnica	Spalnica	Garderoba
Jedilnica	Dnevna soba	Dnevna soba	Dnevna soba	Kuhinja
Kuhinja	Dnevna soba	Dnevna soba	Jedilnica	Jedilnica
Dnevna soba	Spalnica	Spalnica	Spalnica	Jedilnica

Tabela 4.3: Najpogostejše zamenjave po prostorih za vsako mrežo.



(a) Deeplab_1fc

(b) Deeplab_2fc

(c) Deeplab_3fc

(d) PlacesNet

Slika 4.11: Matrike zamenjav.

Poglavje 5

Sklep

V diplomskem delu smo obravnavali problem razpoznavanja prostorov. Naloga je zahtevna tudi za človeškega označevalca. Pogosto meje med prostori niso točno določene, kar predstavlja precejšnjo raven šuma v podatkih, včasih pa so si prostori tako podobni, da jih tudi človek najverjetneje ne bi znal pravilno uvrstiti. Z razvojem avtonomnih vozil ter hišnih robotov se bo najverjetneje vedno bolj kazala potreba po zmožnosti natančnega razpoznavanja prostorov. Problema smo se v nalogi lotili s pomočjo semantične segmentacije, ki sliko klasificira v razrede na nivoju slikovnih točk, da bi s tem dosegli večjo točnost na slikah, ki prikazujejo več kot en prostor.

Na podlagi obstoječe konvolucijske nevronske mreže za semantično segmentacijo smo izdelali tri nove modele, ki se med seboj razlikujejo po številu polno povezanih nivojev. Za namen učenja in testiranja smo obstoječi podatkovni zbirki spremenili oznake za namen segmentacije in dodali slike z več prostori. Naučene modele smo primerjali z mrežo PlacesNet [7], ki celotno sliko uvrsti v en prostor. Natančnost segmentacije smo ovrednotili s povprečno vrednostjo količnika med presekom in unijo označeno z mIOU, pravilnost detekcije prostorov pa s preciznostjo, priklicem in mero F. Referenčna metoda dosega le 48% točnost segmentacije, in 58% točnost detekcije dominantnih prostorov. Vsi modeli, razviti v tej nalogi, v vseh merah dosegajo boljše rezultate. Najboljše rezulate dosega mreža Deeplab_2fc, ki dosega

natančnost segmentacije 0,845, točnost klasifikacije pa 0,89022. Od mreže Deeplab_3fc se v meri mIOU razlikuje na tretji decimalki, v meri F pa na četrti, iz česar lahko zaključimo, da sta mreži približno enako uspešni. Mreža Deeplab_1fc je le za slaba dva odstotka slabša v problemu prepoznavanja prostorov v sliki, vendar je manj uspešna z vidika lokalizacije teh prostorov, saj je njena vrednost mIOU kar slabih deset odstotkov nižja. Najboljši izmed modelov za 20% izboljšuje uspešnost detekcije prostorov referenčne metode zaradi boljšega obvladovanja rotiranih in skaliranih slik. Z našo metodo je mogoče doseči rezultate, ki z vidika točnosti segmentacije za slabih 40% izboljšajo trenutno najuspešnejše metode, kadar slike vsebujejo več kot en prostor.

5.1 Možne izboljšave in nadaljnje delo

Rešitev omogoča veliko možnosti za izboljšavo ter daje veliko idej za nadaljnje delo. Prva takšna izboljšava bi bila, kot že omenjeno v jedru naloge, uporaba navzkrižne validacije za določanje optimalnejših hiperparametrov mreže, ki imajo velik vpliv na točnost in splošnost naučenega modela.

Pri evaluaciji se je kljub večkratnem pregledu podatkovne zbirke pokazalo nekaj napačnih oziroma slabih segmentacij slik iz osnovne zbirke in nesmiselnih umetno generiranih slik. Ker so rezultati lahko dobri samo toliko, kolikor je dobra podatkovna zbirka, na kateri smo mrežo učili, bi bilo potrebno zbirko popraviti in dopolniti. Smiselna bi bila tudi drugačna oblika zapisa pravilnih anotacij, saj trenutno rotiranje in spreminjanje velikosti slik z večimi prostori ni mogoče zaradi interpolacije. Vrednosti so namreč cela števila med 0 in 8, interpolacija pa vanje vnese nove vrednosti. Tako bi naprimer slika na meji med delom z oznako 5 in delom z oznako 7 dobila oznako 6, ki je popolnoma nesmiselna.

Za izboljšanje natančnosti segmentacije bi lahko uporabili CRF [25], po zgledu avtorjev mreže Deeplab [16], ali pa jih predelali glede na potrebe mreže. V vsakem primeru bi lahko s pomočjo informacije o oznakah sosednih

točk programsko določili smiselnost določene oznake in jo spremenili, če bi bila ustreznejša kakšna druga.

Literatura

[1] A. Quattoni and A. Torralba, "Recognizing indoor scenes," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 413–420, 2009.

[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.

[3] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[4] B. Ayers and M. Boutell, "Home interior classification using sift keypoint histograms," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, June 2007.

[5] L. jia Li, H. Su, L. Fei-fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds.), pp. 1378–1386, Curran Associates, Inc., 2010.

[6] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo.," in *CVPR*, pp. 3485–3492, IEEE Computer Society, 2010.

[7] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 487–495, Curran Associates, Inc., 2014.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *Arxiv 2016*, 2016.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[10] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," *CoRR*, vol. abs/1403.1840, 2014.

[11] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015.

[12] S. N. Parizi, A. Vedaldi, A. Zisserman, and P. F. Felzenszwalb, "Automatic discovery and optimization of parts for image classification," *CoRR*, vol. abs/1412.6598, 2014.

[13] P. Uršič, R. Mandeljc, A. Leonardis, and M. Kristan, "Part-based room categorization for household service robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2287–2294, May 2016.

[14] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba., "Semantic understanding of scenes through ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[16] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.

[17] "Neuron. Wikipedia." Dosegljivo: `https://en.wikipedia.org/wiki/Neuron`. [Dostopano: 9. 8. 2016].

[18] "Cs231n: Convolutional Neural Networks for visual recognition." Dosegljivo: `https://cs231n.github.io/`. [Dostopano: 10. 8. 2016].

[19] "Activation function. Wikipedia." Dosegljivo: `https://en.wikipedia.org/wiki/Activation_function`. [Dostopano: 10. 8. 2016].

[20] "Loss functions for classification. Wikipedia." Dosegljivo: `https://en.wikipedia.org/wiki/Loss_functions_for_classification`. [Dostopano: 10. 8. 2016].

[21] M. Nielsen, "Neural networks and deep learning." Dosegljivo: `http://neuralnetworksanddeeplearning.com/chap3.html`, Januar 2016. [Dostopano: 1. 9. 2016].

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] G. Csurka, D. Larlus, and F. Perronnin, "Csurka, larlus, perronnin: Evaluation of semantic segmentation what is a good evaluation measure for semantic segmentation?," 2013.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CVPR (to appear)*, Nov. 2015.

[25] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012.